
Physical Construction Toys for Rapid Sketching of Tangible User Interfaces

Kristian Gohlke

Bauhaus-Universität Weimar
Geschwister-Scholl-Str. 7, 99423 Weimar
kristian.gohlke@uni-weimar.de

Michael Hlatky

Native Instruments GmbH
Schlesische Str. 29-30, 10997 Berlin
michael.hlatky@native-instruments.de

Bram de Jong

Native Instruments GmbH
Schlesische Str. 29-30, 10997 Berlin
bram.dejong@native-instruments.de

Abstract

Physical construction toys such as *Lego* or similar brick-based systems provide a rich space for tangible interactions. In this paper we explore their potential as a readily available prototyping platform for Tangible User Interfaces. The system enables interaction designers and users to create functional control layouts, tangible experience prototypes, custom interface elements, and grid based interactions from *Lego* bricks. The position, color, orientation and shape of the markerless bricks are tracked using a camera. We have conducted an initial exploration of this design space through an exemplary case study of a tangible interface for music creation: users can construct interactive interface elements such as tangible grids for the arrangement of drum-patterns and different types of continuous controllers for parameter adjustments. The system is currently a work in progress; ongoing efforts are aimed at a more thorough evaluation.

Author Keywords

Tangible Interaction; TUI; Interaction Design; Lego; Prototyping; Music Controllers; HCI; Physical Sketching; Sequencer; Sliding Patterns; Sliding Puzzle.

ACM Classification Keywords

H.5.2. User Interfaces: *Prototyping*. H.5.5 Sound and Music Computing: *Methodologies and techniques*.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.

Copyright is held by the owner/author(s).

TEI '15, Jan 16-19 2015, Stanford, CA, USA

ACM 978-1-4503-3305-4/15/01.

<http://dx.doi.org/10.1145/2677199.2687900>

Introduction

This paper describes the work-in-progress prototype of a system that enables interaction designers to rapidly create tangible interface layouts and novel interface elements using *Lego* bricks as a readily available construction system. Such experience prototypes [3] can serve as functional interface sketches for rapid evaluation of interface ideas and workflows and can also provide a basis for decision making in user-centered, participatory design processes. We assume that the use of *Lego* as a familiar construction material can increase the user acceptance during the evaluation of early prototypes and might reduce additional overhead for learning [see: 7]. Different *interface widgets* and tangible tokens can be built from *Lego* bricks and placed on a translucent *Lego* baseplate. A camera mounted underneath the baseplate is used to track the bricks. The visual tracking does not require additional visual markers. Currently the *color, position, orientation, size* and *shape* of each object being placed on the baseplate are analyzed. Widgets of the same type and the same color are distinguished by their position. The camera-image is processed in software and displayed on a separate screen that also serves as a back-end for configuring the tangible widgets. The user can assign functionality to each physical interface widget and token by defining active areas through an additional graphical user interface (GUI). The extracted information is mapped to control messages, currently OpenSoundControl (OSC) and MIDI – as our initial case studies are focused on prototyping user interfaces for audio applications. However, the system is prepared to generate other events and messages such as keystrokes, mouse events, HID and serial messages to enable the control of a variety of existing or novel software and hardware.

Related Work

Previous work from members of our research group [2] has explored the use of ubiquitous, readily available household items, such as string, wire, tape, cardboard and similar to identify and conceive ways to create and interact with novel musical instruments.

An audio sequencer that uses a tangible grid and colored tokens has been proposed by Hesse et al. [8]. Their system allows for the creation of drum loops by placing chewing-gum balls on a fixed grid of holes with four rows and sixteen columns, representing one bar of music. The *Tactus* [14] is a musical interface that enables the use of everyday items, such as board game tokens or drawings, to create rhythmic patterns on grids that are tracked by a camera. A marker-based, low-cost approach for evaluation and prototyping of TUIs has been proposed by Costanza et al. [4]. Their system called *d-touch* allows users to work with printable optical markers to create their own tangibles. The *d-touch* system works similar to the *reactivision* framework [10] and has also been demonstrated as a musical interface. An extensive overview of tangible interfaces for music performance and production is available online [11].

The use of *Lego* bricks for creating rhythmic patterns has been demonstrated through a variety of *Lego*-based sequencers that use unmodified bricks such as the *Brickquencer* [23], *BeatBricks* [22], the *SoundMachine* [24] or the *Poco* [25], using *Lego Duplo* bricks. A second category of sequencers uses modified bricks such as the *Lego Sequencer MR2* [1] and similar [21]. The *Stekgreif* [9] is a modular MIDI-controller that allows for the creation of functional music controller layouts from modified *Lego Duplo* bricks with integrated electronics. The recently released *Lego Fusion* [12]

consists of a special baseplate that can be tracked visually through an application running on a tablet-computer or smartphone. *Lego* facades with a maximum size of 16x16 bricks can be constructed and imported into an application where they are extruded into virtual 3D-models, serving as content for screen-based mini games. The commercially available *Osmo* [18] uses a tablet-computer with a custom stand and a small mirror attachment that slides over the tablet's built-in camera. The mirror redirects the camera viewing angle in order to detect objects placed on surfaces in front of the device. The objects can be used to interact with a variety of games and other applications. *Palette* [19] is a modular system of physical controllers including a button module, a rotary knob, and a slider that can be freely arranged and connected to a computer through an additional module. Configuration software allows users to create custom controller mappings, to interact with a range of different applications. However, the interactive capabilities are limited by the number of modules available in a given context and new modules cannot be easily created by the user. The *SLAP Widgets* [26] add tactile controls to multi-touch tables. A set of passive, translucent buttons, knobs and sliders can be positioned on an interactive surface and are tracked using markers. Visual feedback is provided by rear projection.

Interface Widgets

Using our system, currently the following interface widgets can be constructed and used for prototyping: *rotary controls*, *linear sliders*, *x/y pads*, *switches* and *grid areas* [see Fig. 1]. *Rotary controls* are constructed from a standard *Lego* brick (e.g. 2x3 brick) that is attached on top of a rotary *Lego* plate (e.g. a 4x4 or

2x2 rotary plate) and then mounted on the translucent baseplate. The overlapping part of the brick can then be tracked using a camera. *Linear sliders* and *x/y pads* are constructed by setting out an area on the baseplate that is delimited by surrounding bricks. One or several *Lego* bricks (e.g. 2x2 brick) that are placed inside the defined area serve as tokens to generate x/y coordinates. Linear sliders are constructed by the same principle but have to be built in such a way that movement is restricted to a single axis. The area of a slider or an x/y pad can also be covered with studless, transparent *Lego* tiles, so that the user can slide tokens across the surface. Alternatively, when no transparent tiles are available, flat colored *Lego* tiles (e.g. 2x2) without studs can also serve as movable tokens, if placed on the surface with their top side facing downwards. Multiple x/y coordinates can be generated simultaneously by using different tokens that are distinguished by color. *On/off switches* are implemented based on the presence or absence of a token (e.g. 2x1 brick) at a user-defined position. The token can either be affixed to the baseplate, or moved into position similar to a small linear slider. Interactive *Grid areas* are essentially rows and columns of multiple switches that can cover larger sections of the interface.

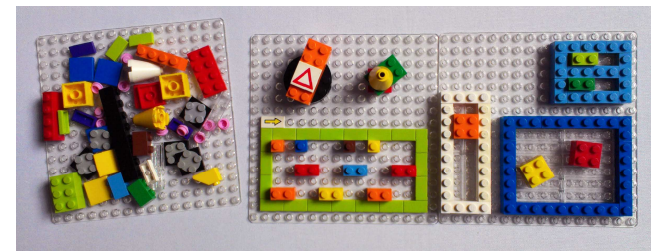


Figure 1. Right: examples of widgets, e.g. rotary controls, a grid area, a linear slider, switches, and an x/y pad. Left: exemplary collection of the brick types used to construct such widgets.

User Defined Control Layouts

The process of creating control layouts is straightforward. First, the user builds interface widgets based on the examples described above and places them on the interactive surface. A live camera image, shot from underneath the surface is combined with a graphical overlay and displayed on a separate screen that serves as a GUI backend. This backend allows the user to define regions of interest inside the image, declare them as active tracking areas and assign the desired control mappings. Visual feedback is given to indicate whether a physical widget placed on an active area could be recognized or if the widget needs to be modified to improve the tracking. The use of a projector adjacent to the camera would also allow for an integration of a GUI with the interactive surface. However the approach of using a separate screen was sufficient to test our initial assumptions, since the current GUI served solely as a configuration back-end at design time, but is not required when interacting with the tangible widgets.

Image Analysis Engine

In our current setup, a high-definition camera is mounted underneath a translucent *Lego* baseplate measuring 48x48 studs. The image processing is based on the OpenCV [17] library. Perspective correction, using four manually adjustable corner points is applied to the image to remove skewing and compensate for the camera angle. The corrected image is analyzed, using custom algorithms, to deduce the properties of objects being placed on the baseplate. Currently the system is capable of detecting the color, the size, the contours, the orientation and the x/y coordinates of the footprint of objects being placed on the surface. Robust color tracking in HSV color space is achieved through

an adaptive color matching algorithm that can be calibrated with a selection of the available brick colors. Additional colors can be defined to distinguish between bricks and the hands of the user. The information derived from the image is then further abstracted and mapped to parameters corresponding to the type of interface widget that the user assigned to a given image area. The underlying data-structure for *linear slider widgets* holds values that correspond to the position of the sliding token within the widget area (i.e. the actual value used to adjust parameters) and the color value of the token. *Rotary control widgets* contain the same information but here the orientation angle corresponds to the parameter value. The data-structure for areas designated as *x/y pads* holds information about the x/y position of each token relative to the widget area, the color value for each token, the number of tokens and their orientation angle. *Switch widgets* contain a binary on/off information (token present or not present) and the color value of the token. *Grid areas* hold the same information as individual switches, but for each cell of the grid. The values of switches and grid areas can be quantized to match the base grid of the *Lego* plate. The individual values for each detected widget are then scaled and mapped to generate the control events and messages as specified by the user.

Case Study: A Tangible Music Controller

We have built a tangible rhythm sequencer [see Fig. 2] as a case study to demonstrate some of the features of a markerless system for sketching TUIs using unmodified *Lego* bricks. The prototypes were built during the *Midihack Stockholm 2014* [15] and the *MusicHackDay Barcelona 2014* [16]. Informal observations were made at these occasions with groups of users (most of them having a background in audio

production, music, and/or software development) to identify the potential of our approach as a platform for prototyping TUIs for musical interactions. Our current prototypes can be used create rhythmic drum patterns by arranging colored *Lego*-bricks on a transparent baseplate that is illuminated from below. The case study combines different interface elements. Beyond the user-manipulated physical arrangement of the interface elements, no additional visual feedback is provided.



Figure 2. Case study: a tangible rhythm sequencer with parameter controls. The translucent top surface of the table is covered with different types of widgets. A high definition camera and illumination are mounted underneath.

Variations: Sliding Patterns and Sushi Plates

Inspired by initial observations of people constructing their own physical tools from *Lego* bricks to prepare and share drum patterns, the 48x48 stud baseplate was replaced by eight smaller plates measuring 16x16 studs and an empty area of the same size. Similar to a sliding puzzle toy, the eight plates can be moved on the surface horizontally and vertically [13]. This allows users to shift entire patterns of tokens into the visually tracked area, enabling instant playback and shifting of pre-built drum patterns. The sliding patterns also work in conjunction with regular interface elements by locking individual baseplates to the edge of the surface.

Our system is not restricted to tracking *Lego* bricks. We have demonstrated this through another variation of a tangible music sequencer that uses differently colored sushi plates, running on a sushi conveyor belt as a TUI in an actual live performance setting [20].

Conclusion and Outlook

We presented a work-in-progress system for the rapid creation ("*physical sketching*") of functional, tangible controller layouts for music production using *Lego* bricks. The controller layouts can be quickly customized and evaluated. Ongoing work is aimed at extending the prototyping capabilities by providing users with more flexibility and customization, to support them in conceiving entirely new interface elements. For example by enabling users to teach the system to detect custom combinations of bricks, similar to a proposed method for optical tracking that replaces pre-printed markers with hand drawn sketches [5]. This approach could be further extended to work with depth cameras similar to the approach described by Gupta et al. [6] to add more volume to the interactive space.

Acknowledgements

The research proposed in this paper is performed in the *GiantSteps* project, which has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement n° 610591.

References

1. Akai, Y. LegoSequencer MR2. yoshiakai.com/2008/Lego%20Sequencer01.html (accessed: July 25th 2014)
2. Andersen, K. Using Props to Explore Design Futures: Making New Instruments. *Proceedings of ACM CHI '14*. ACM Press (2014).
3. Buchenau, M., & Fulton Suri, J. Experience prototyping. *Proc. ACM DIS '00*, ACM Press (2000).
4. Costanza, E., et al. Ubicomp to the masses: a large-scale study of two tangible interfaces for download. *Proc. ACM Ubicomp '10*. ACM Press (2014).
5. Costanza, E. & Huang, J. Designable Visual Markers. *Proc. ACM CHI '09*. ACM Press (2009).
6. Gupta, A., et al. DuploTrack: a real-time system for authoring and guiding Duplo block assembly. *Proc. UIST '12*. ACM Press (2012).
7. Harrison, C., et al. TouchTools: leveraging familiarity and skill with physical tools to augment touch interaction. *Proc. ACM CHI '14*. ACM Press (2014).
8. Hesse, A., & McDiarmid, A. The Bubblegum Sequencer. *Proc. ACM CHI 2008*. ACM Press (2008). www.backin.de/gumball/ (acc.: July 25th 2014)
9. Homann, G., Tzschentke, D., Weidenhaupt, C. & Zentis, P. Stekgreif und an.schlag. *Proc. Mensch & Computer 2008, Berlin*. Logos Verlag.
10. Kaltenbrunner, M., & Bencina, R. reacTIVision: a computer-vision framework for table-based tangible interaction. *Proc. ACM TEI '07*. ACM Press (2007).
11. Kaltenbrunner, M., Tangible Music. modin.yuri.at/tangibles/ (acc.: July 25th 2014).
12. Lego Fusion. www.lego.com/en-us/fusion/ (acc.: July 25th 2014)
13. LegoTechno 2.0 - Sliding Puzzle Sequencer www.youtube.com/watch?v=ei9jyGvRgvs (acc.: November 11th 2014)
14. Mann, Y., Lubow, J., & Freed, A. The Tactus: A Tangible, Rhythmic Grid Interface Using Found-Objects. *Proc. Nime 2009*.
15. MidiHack 2014. www.midihack.com (acc.: July 25th 2014)
16. Music Hack Day Barcelona 2014. new.musichackday.org/2014/barcelona/ (acc: July 25th 2014)
17. OpenCV. www.opencv.org (acc.: July 25th 2014)
18. Osmo. www.playosmo.com (acc: July 25th 2014)
19. Palette. www.palettegear.com (acc.: July 25th 2014)
20. RBMA Presents: Sushi Sequencer (Making Of) www.youtube.com/watch?v=rZFchIXL7nk (acc.: November 11th 2014)
21. Rosenbaum, E., Lego Sequencer. www.youtube.com/watch?v=eL7F58KantE (acc.: July 25th 2014)
22. Superquadratic - Beat Bricks. www.vimeo.com/45026119 (acc.: July 25th 2014)
23. The Brickquencer. www.fleg.de/art/brickquencer (acc.: July 25th 2014)
24. The SoundMachine. www.thinkbricks.net/?p=638 (acc.: July 25th 2014).
25. Wagner, T. Poco Sequencer. www.thomaswagner.me/5poco.html (acc.: July 25th 2014)
26. Weiss, M., et al. SLAP widgets: bridging the gap between virtual and physical controls on tabletops. *Proc. ACM CHI '09*. ACM Press (2009).